Figs. 10 and 11 are schematic representations of two memory arrays for the Flash EEprom embodiments of this invention;

Fig. 12 contains Table I which shows voltage conditions for all operational modes for the array of Fig. 10; and

Fig. 13 contains Table II which shows example voltage conditions for all operational modes for the virtual ground array of Fig. 11.

At page 32, line 16, please insert the following new paragraphs:

Fig. 9 outlines the main steps in the sequence of the new erase algorithm. Assume that a block array of mxn memory cells is to be fully erased (Flash erase) to state "3" (highest conductivity and lowest V_{T1} state). Certain parameters are established in conjunction with the erase algorithm. They are listed in Fig. 9: V_I is the erase voltage of the first erase pulse. V_I is lower by perhaps 5 volts from the erase voltage required to erase a virgin device to state "3" in a one second erase pulse. t is chosen to be approximately 1/10 th of the time required to fully erase a virgin device to state "3". Typically, V_I may be between 10 and 20 volts while t may be between 10 and 100 milliseconds. The algorithm assumes that a certain small number, X, of bad bits can be tolerated by the system (through for example error detection and correction schemes implemented at the system level. If no error detection and correction is implemented then X =0). These would be bits which may have a shorted or leaky tunnel dielectric which prevents them from being erased even after a very long erase pulse. To avoid excessive erasing the total number of erase pulses in a complete block erase cycling can be limited to a preset number, n_{max}. ΔV is the voltage by which each successive erase pulse is incremented. Typically, ΔV is in the range between 0.25V and 1.0V. For example, if V_I =15.0V and ΔV=1.0V, then the seventh erase pulse will be of magnitude V_{ERASE} =21.0V and duration t. A cell is considered to be fully erased when its read conductance is greater than $I_{"3"}$. The number S of complete erase cyclings experienced by each block is an important information at the system level. If S is known for each block then a block can

be replaced automatically with a new redundant block once S reaches $1x10^6$ (or any other set number) of program/erase cycles. S is set at zero initially, and is incremented by one for each complete block erase multiple pulse cycle. The value of S at any one time can be stored by using for example twenty bits (2^{20} equals approximately $1x10^6$) in each block. That way each block carries its own endurance history. Alternatively the S value can be stored off chip as part of the system.

The sequence for a complete erase cycle of the new algorithm is as follows (see Fig. 9):

1. Read S. This value can be stored in a register file. (This step can be omitted if S is not expected to approach the endurance limit during the operating lifetime of the device).

1a. Apply a first erase pulse with $V_{ERASE} = V_1 + n\Delta V$, n=0, pulse duration=t. This pulse (and the next few successive pulses) is insufficient to fully erase all memory cells, but it serves to reduce the charge Q on programmed cells at a relatively low erase field stress, i.e., it is equivalent to a "conditioning" pulse.

D2 Cont.

1b. Read a sparse pattern of cells in the array. A diagonal read pattern for example will read m+n cells (rather than mxn cells for a complete read) and will have at least one cell from each row and one cell from each column in the array. The number N of cells not fully erased to state "3" is counted and compared with X.

1c. If N is greater than x (array not adequately erased) a second erase pulse is applied of magnitude greater by ΔV than the magnitude of the first pulse, with the same pulse duration, t. Read diagonal cells, count N.

This cycling of erase pulse/read/increment erase pulse is continued until either $N \leq X$ or the number n of erase pulses exceed n_{max}. The first one of these two conditions to occur leads to a final erase pulse.

2a. The final erase pulse is applied to assure that the array is solidly and fully erased. The magnitude of V_{ERASE} can be the same as in the previous pulse or higher by another increment ΔV. The duration can be between 1t and 5t.

2b. 100% of the array is read. The number N of cells not fully erased is counted. If N is less than or equal to X, then the erase pulsing is completed at this point.

2c. If N is greater than X, then address locations of the N unerased bits are generated, possibly for substitution with redundant good bits at the system level. If N is significantly larger than X (for example, if N represents perhaps 5% of the total number of cells), then a flag may be raised, to indicate to the user that the array may have reached its endurance end of life.

2d. Erase pulsing is ended.

3a. S is incremented by one and the new S is stored for future reference. This step is optional. The new S can be stored either by writing it into the newly erased block or off chip in a separate register file.

D2 Cont.

3b. The erase cycle is ended. The complete cycle is expected to be completed with between 10 to 20 erase pulses and to last a total of approximately one second.

The new algorithm has the following advantages:

(a) No cell in the array experiences the peak electric field stress. By the time V_{ERASE} is incremented to a relatively high voltage any charge Q on the floating gates has already been removed in previous lower voltage erase pulses.

(b) The total erase time is significantly shorter than the fixed V_{ERASE} pulse of the prior art. Virgin devices see the minimum pulse duration necessary to erase. Devices which have undergone more than $1x10^4$ cycles require only several more ΔV voltage increments to overcome dielectric trapped charge, which only adds several hundred milliseconds to their total erase time.

(c) The window closure on the erase side (see curve (b) in FIG. 11d of U.S. patent number 5,095,344) is avoided indefinitely (until the device experiences failure by a catastrophic breakdown) because V_{ERASE} is simply incremented until the device is erased properly to state "3". Thus, the new erase algorithm preserves the full memory window.

Flash EEprom Memory Array Implementations

The Flash EEprom cells of this invention can be implemented in dense memory arrays in several diffcrent array architectures. The first architecture, shown in Fig. 10, is the one commonly used in the industry for Eprom arrays. The 3x2 array of Fig. 10 shows two rows and three columns of Flash EEprom transistors. Transistors T10, T11, T12 along the first row share a common control gate (word line) and a common source S. Each transistor in the row has its own drain D connected to a column bit line which is shared with the drains of all other transistors in the same column. The floating gates of all transistors are adjacent their drains, away from their sources. Erase lines are shown running in the bit line direction (can also run in the word line direction), with each erase line coupled (through the erase dielectric) to the floating gates of the transistors to the left and to the right of the erase line. The voltage conditions for the different modes of operation are shown in Table I (Fig. 12) for the selected cell as well as for unselected cells sharing either the same row (word line) or the same column (bit line). During block erase of all the cells in the array, all erase lines are brought high. However, it is also possible to erase only scctors of the array by taking V_{ERASE} high for pairs of erase gates only in these sectors, keeping all other erase lines at 0V.

D2 cnt.

A second Flash EEprom memory array architecture which lends itself to better packing density than the array of Fig. 10 is known as the virtual ground array (for a detailed description of this array architecture, see the Harari patent referenced herein). A topological view of such an array of cells is provided in Figs. 6a, 7a, 8a and 9a of U.S. patent number 5,095,344. A schematic representation of a 2x2 virtual ground memory array corresponding to the array of Fig. 6a of U.S. patent number 5,095,344 is shown in Fig. 11. In a virtual ground array, the source and drain regions are used interchangeably. For example, diffusion 502 is used as the drain of transistor 600a and as the source of transistor 600b. The term "virtual ground comes from the fact that the ground supply applied to the source is decoded rather than hard-wired. This decoding allows the source to be used interchangeably as ground line or drain. The operating conditions in the virtual ground array are given in Table II (Fig. 13). They are essentially the same as that for the

standard architecture array, except that all source and drain columns of unselected cells are left floating during programming to prevent accidental program disturbance. During reading all columns are pulled up to a low voltage (about 1.5V) and the selected cell alone has its source diffusion pulled down close to ground potential so that its current can be sensed.

D2 Cont.

The array can be erased in a block, or in entire rows by decoding the erase voltage to the corresponding erase lines.

Fig. 12 provides Table I which shows voltage conditions for the operational modes of the array in Fig. 10.

Fig. 13 provides Table II which shows example voltage conditions for the operational modes for the virtual ground array of Fig. 11A.

IN THE DRAWINGS:

Please add new drawing figures 9-13.